An Efficient Integrated Scheduling Method for Twin-Cluster Tools with Buffer Space Configuration

Yiming Lai, Chunrong Pan, Wenqing Xiong, Jinchao Fu, Yu Cui and Hao Zhou

November 2, 2023

An Efficient Integrated Scheduling Method for Twin-Cluster Tools with Buffer Space Configuration

YiMing Lai
School of Mechanical and Electrical Engineering
Jiangxi University of Science and Technology
Ganzhou, China
ymlai_001@163.com

ChunRong Pan
School of Mechanical and Electrical Engineering
Jiangxi University of Science and Technology
Ganzhou, China
crpan@jxust.edu.cn

WenQing Xiong
School of Mechanical and Electrical Engineering
Jiangxi University of Science and Technology
Ganzhou, China
wqxiong_study@163.com

JinChao Fu
Zhuhai IKAS Smart Technology Company
Zhuhai, China
fujinchaojy@163.com

Yu Cui
School of Mechanical and Electrical Engineering
Jiangxi University of Science and Technology
Ganzhou, China
cuiyu9811@163.com

Hao Zhou
School of Mechanical and Electrical Engineering
Jiangxi University of Science and Technology
Ganzhou, China
zhouhaostudy@163.com

***Abstract*—For improving the efficiency and reducing the processing load of a single tool, as novel equipment that have multiple processing routes, twin-cluster tools are widely used in wafer manufacturing. Periodic schedule has an important role in controlling wafers quality. It is meaningful to schedule a twin-cluster tool such that, processing routes in the tool work periodically. To achieve this goal, the workload of each module is carefully analyzed, leading to that the optimal cycle time of the system can be obtained. Then, a mathematical programming model is presented to make sure twin-cluster tools achieve the minimal cycle time. Further, a novel algorithm is established so as to obtain the robot waiting time. Furthermore, several sufficient conditions are proposed to guarantee that twin-cluster tools achieve the optimal cycle time. Finally, numerous experiments are conducted to verify the accuracy of the conclusion with different conditions.**

Keywords—twin-cluster tool, semiconductor manufacturing, periodic scheduling, mathematical programming model

I. Introduction

Cluster tools are widely used in semiconductor manufacturing for wafer fabrication. A cluster tool contains several *processing chambers* (PCs), a transport module, and two *loadlocks* (LLs). As shown in Fig. 1, with one or two arms, tools can be classified as single-arm cluster tools and dual-arm cluster tools. From a LL, the raw wafer enters to the tool and is transported to each PC for processing through the transport module. Many studies are conducted in scheduling cluster tools [1]–[6]. For a cluster tool, if the cycle time equals to the robot task time, the tool is transport-bound, otherwise, it is process-bound. For a process-bound tool, a backward strategy is proved that is optimal for single-arm cluster tools [7], and a swap strategy is better for dual-arm cluster tools [8].

In recent years, the time needed for wafer processing in a PC is increasing to satisfy the complex processing requirement. More and more PCs are installed on the tool to reduce the workload, until the maximum number of PCs in a tool is reached. To further reduce the workload and improve the tool space utilization efficiency, two or more individual cluster tools are connected through *Buffer Module* (BM) to form a

This work was funded by Distinguished Professor Program of Jinggang Scholars in institutions of higher learning, Jiangxi Province.

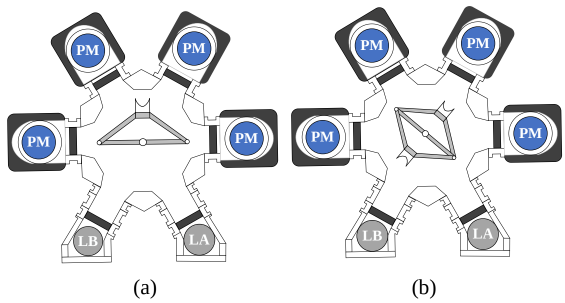

Figure 1. Cluster tools: (a) A single-arm tool; (b) A dual-arm tool

semiconductor tool. Such a tool is called multi-cluster tool and widely used in recent years [10].

Many studies have been made for the multi-cluster tool scheduling. Based on the event graph and network models, Ding *et al*. [10] find a feasible robot sequence and calculate the corresponding cycle time. Yi *et al*. [11] suggest to disassemble a multi-cluster tool into several individual cluster tools, and an existed individual cluster tool scheduling method is applied to each one. Chan *et al*. [12] present a resource-based model to analyze the minimal cycle time of twin-cluster tools and propose an algorithm to find the optimal schedule. Zhu *et al*. [13]–[15] point out that, the stability and simplicity of a system operation are guaranteed in the one-wafer schedule. With one-wafer schedule, based on the petri net, they analyze the schedulability of single- and dual-arm multi-cluster tools and obtain the cycle time by scheduling robot waiting times.

A twin-cluster tool is a kind of two-cluster tool. Different from typical multi-cluster tools, twin-cluster tools feature two cluster tool units possessing identical processing technologies such that, wafers can choose one unit to complete all processes without the need that must be transported to all units. It is also allowed that one wafer is processed in both tool units for different processes. However, since the robot cooperation between two units becomes more frequent in this mode, there are more risks to deadlock. As a result, the processing mode that wafers are processed in both units for partial processes is rarely implemented in practical production. Meanwhile, periodic schedule has an important role in controlling wafers

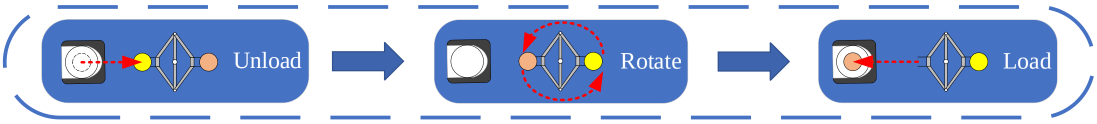

Figure 2. A swap strategy

quality. Therefore, this paper aims to find a feasible schedule method for process-bound twin-cluster tools such that both tool units can periodically complete wafers efficiently.

This paper is organized as follows. In Section II, we describe the configuration and manufacturing requirement of twin-cluster tools, and analyze the workload of each component. In Section III, we present an efficient mathematical programming model and the corresponding algorithm. In Section IV, experiments are conducted to verify the accuracy of the conclusion. Finally, we summarize our findings in Section V.

II. Scheduling Analysis

A. Configuration of a Twin-Cluster Tool and Manufacturing Requirements

A twin-cluster tool studied in this paper (see Fig. 3) consists of two cluster tool units through a BM. The cluster tool unit connected with LLs and the BM is called *cluster tool 1* (CT1), and the other is called *cluster tool 2* (CT2). For a cluster tool unit, there are several PCs serving for Step 1 or Step 2. Let $\mathbf{N}_n = \{0, 1, 2, ..., n\}$, and $\mathbf{N}_n^+ = \mathbf{N}_n / 0$. $m = [(m_{11}, m_{12}), (m_{21}, m_{22})]$ is used to denote the *wafer flow patten* (WFP) of a twin-cluster tool, where m_{ij} denotes the number of PCs at Step i in CTj, $i, j\in\mathbf{N}_2^+$. CT1 and CT2 both have a dual-arm robot, which are denoted by R_1 and R_2, respectively. A row wafer in the twin-cluster tool must be processed once in both Step 1 and Step 2, respectively.

For CTj, let $U_{i\text{-}j}$ denote the unloading task that robot unloads a wafer from a PM of Step i, $L_{i\text{-}j}$ denote the loading task that robot loads a wafer into a PM of Step i, $T_{i\text{-}j}$ denote the rotating task that robot rotates at Step i with a wafer, and $M_{il\text{-}j}$ is used to denote the moving task that R_j moves form Step i to Step l, $i, l\in\mathbf{N}_2, j\in\mathbf{N}_2^+$. Note that Step 0 and b represent LLs and the BM. Robot tasks take time. Let ρ_i denote the time required for Step i to process, and α_j and ν_j denote the time required for R_j to unload/load and rotate, respectively, $i, j\in\mathbf{N}_2^+$. Let Π_{ij} denote the workload of a PC at Step i in CTj, $i, j\in\mathbf{N}_2^+$.

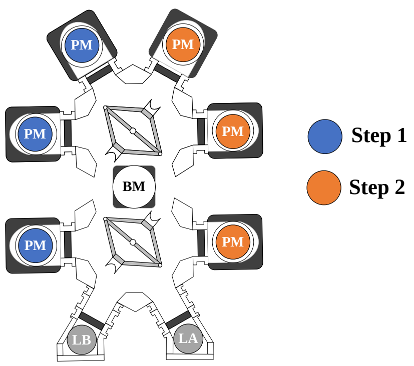

Figure 3. A twin-cluster tool

B. Workload Analysis

For a process-bound dual-arm cluster tool, a swap strategy is optimal in terms of the cycle time. As shown in Fig. 2, for Step i, with the swap strategy, until a wafer in the PC is completed, the robot comes to the PC and unload the wafer, then rotate and load another raw wafer into there. Above-mentioned task sequence can be denoted by $\langle U_i \rightarrow T_i \rightarrow L_i\rangle$. For a processing route, after the robot performs the sequence for every step, the modules of each step in the route return to the initial state. This is called a robot task cycle. Notice that, in a robot task cycle, after the robot performs $\langle U_i \rightarrow T_i \rightarrow L_i\rangle$, a wafer starts to be processed in a PC at Step i. Until the robot performs $\langle U_i \rightarrow T_i \rightarrow L_i\rangle$ again, the processed wafer is unloaded and another raw wafer is loaded for processing. It implies that there is a wafer completion cycle at the PC including $\langle U_i \rightarrow T_i \rightarrow L_i\rangle$, and a processing activity. This consumes times $\rho_i + 4\alpha + 3\nu$ that is called the workload of the PC. Since there are m_{ij} PC serving for Step i in CTj, then we have

$$\Pi_{ij} = (\rho_i + 2 \times \alpha_j + \nu_j) / m_{ij},\ i, j\in\mathbf{N}_2^+ \tag{1}$$

With the assumption that each wafer needs to be completed in CT1 or CT2, there are two wafer processing routes in a twin-cluster tool, such as

- *Route* 1: LL → Step 1 (CT1) → Step 2 (CT1) → LL
- *Route* 2: LL → BM → Step 1 (CT2) → Step 2 (CT2) → BM → LL

TABLE I. Robot Task Sequence in One Cycle

	R_1	R_2
Route 1	$M_{01\text{-}1} \rightarrow U_{1\text{-}1} \rightarrow T_{1\text{-}1} \rightarrow L_{1\text{-}1} \rightarrow M_{12\text{-}1} \rightarrow U_{2\text{-}1} \rightarrow T_{2\text{-}1} \rightarrow L_{2\text{-}1} \rightarrow M_{20\text{-}1} \rightarrow U_{0\text{-}1} \rightarrow T_{0\text{-}1} \rightarrow L_{0\text{-}1}$	
Route 2	$M_{0b\text{-}1} \rightarrow U_{b\text{-}1} \rightarrow T_{b\text{-}1} \rightarrow L_{b\text{-}1} \rightarrow M_{b0\text{-}1} \rightarrow U_{0\text{-}1} \rightarrow T_{0\text{-}1} \rightarrow L_{0\text{-}1}$	$M_{b1\text{-}2} \rightarrow U_{1\text{-}2} \rightarrow T_{1\text{-}2} \rightarrow L_{1\text{-}2} \rightarrow M_{12\text{-}2} \rightarrow U_{2\text{-}2} \rightarrow T_{2\text{-}2} \rightarrow L_{2\text{-}1} \rightarrow M_{2b\text{-}2} \rightarrow U_{b\text{-}2} \rightarrow T_{b\text{-}2} \rightarrow L_{b\text{-}2}$

As shown in Table I, when R_1 serves for Route 1, based on the swap strategy, it sequentially comes to Step 1, 2, and 0 to perform swap sequences. In Route 2, R_1 sequentially comes to Step b and 0 to perform swap sequences such that a raw wafer is loaded into the BM and the completed wafer is unloaded, then sent to LL. Meanwhile, R_2 sequentially comes to the PCs of Step 1 and 2 in CT2, and Step b to perform a swap sequence. Let $\theta_{r\text{-}j}$ denote the workload of R_j for Route r in one cycle, $j, r\in\mathbf{N}_2^+$. Then, we have

$$\theta_{1\text{-}1} = 6\nu_1 + 6\alpha_1 \tag{2}$$

$$\theta_{2\text{-}1} = 4\nu_1 + 4\alpha_1 \tag{3}$$

$$\theta_{2\text{-}2} = 6\nu_2 + 6\alpha_2 \tag{4}$$

The workload is the longest time needed for all activities. Based on the above analysis, the workload of Route 1 is $max\ (\Pi_{i1} \mid i\in\mathbf{N}_2^+, \theta_{1\text{-}1})$, and the workload of Route 2 is $max\ (\Pi_{i2} \mid i\in\mathbf{N}_2^+, \theta_{2\text{-}1}, \theta_{2\text{-}2})$. Let Π_r denote the workload of Route r, $r\in\mathbf{N}_2^+$, with the assumption that Route r is process-bound, $j\in\mathbf{N}_2^+$. Then, we have

$$\Pi_r = max\ (\Pi_{ir} \mid i\in\mathbf{N}_2^+),\ r\in\mathbf{N}_2^+ \tag{5}$$

To ensure that R_1 and R_2 serve for Route 1 and 2 without deadlock, we make robots wait after they complete the sequence for Route 1 and 2. Let w_{r-j} denote the robot waiting time of R_j from the time point that robot completes the task sequence for Route r to the time point that robot starts to perform the identical task sequence again, j, $r\in\mathbf{N}_2^+$. Let ϑ_{r-j} denote the one-wafer cycle time of R_j for Route r, j, $r\in\mathbf{N}_2^+$. Then, we have

$$\vartheta_{1\text{-}1} = 6v_1 + 6\alpha_1 + w_{1-1} \tag{6}$$

$$\vartheta_{2\text{-}1} = 4v_1 + 4\alpha_1 + w_{2-1} \tag{7}$$

$$\vartheta_{2\text{-}2} = 6v_2 + 6\alpha_2 + w_{2-2} \tag{8}$$

Notice that, the steady-state process of a processing route is a series process, the production rate is the same to all steps, then, we have

$$\vartheta_{2\text{-}1} = \vartheta_{2\text{-}2} \tag{9}$$

C. Mathematical Programming Model

For a processing route, its workload is the optimal cycle time to complete a wafer. Then, we have the following lemma.

Lemma 1: Suppose that there is a process-bound twin-cluster tool. To ensure the tool achieve the optimal throughput when two wafer processing routes work periodically, (10) and (11) should be satisfied.

$$\vartheta_{1\text{-}1} = \Pi_1 \tag{10}$$

$$\vartheta_{2\text{-}1} = \Pi_2 \tag{11}$$

Proof: According to (5), Π_r is the optimal cycle time of Route r in a process-bound twin-cluster tool. Then, if (10) and (11) are both satisfied, it means that the throughput of each route reaches the maximum value such that, the tool is optimally scheduled. Therefore, the lemma holds. ■

When two processing routes both are scheduled periodically, the system enters to the steady-state. In this time, Route 1 and 2 may have a different cycle time. Let (λ_1, λ_2) denote the sets of the route cycle number, where λ_r indicates the number of Route r cycle required for the system to return to the initial state, $r\in\mathbf{N}_2^+$. Then, we have

$$\lambda_1 \times \vartheta_{1\text{-}1} = \lambda_2 \times \vartheta_{2\text{-}1} \tag{12}$$

According to Table I, R_1 is scheduled for both Route 1 and 2, respectively. Thus, we should properly schedule R_1 to perform the tasks of Route 1 and 2 on time, leading to that Route 1 and 2 periodically work. Let T_{1-k} and t_{1-K} denote the time point that R_1 starts to perform the task sequence of Route 1 and 2 in kth and Kth cycle, $k\in\mathbf{N}_{\lambda_1}^+$, $K\in\mathbf{N}_{\lambda_2}^+$, respectively. Then, we have

Lemma 2: Suppose that there is a process-bound twin-cluster tool. To ensure that Route 1 and 2 both are periodically scheduled in respective cycle time, the following condition should be met.

$$\begin{cases} T_{1-k} \le t_{1-K} - \vartheta_{1\text{-}1},\ T_{1-k} < t_{1-K} \\ t_{1-K} \le T_{1-k} - \vartheta_{2\text{-}2}, t_{1-K} < T_{1-k} \end{cases}, k\in\mathbf{N}_{\lambda_1}^+, K\in\mathbf{N}_{\lambda_2}^+ \tag{13}$$

Proof: According to (6), R_1 performs the sequence of Route 1 every $\vartheta_{1\text{-}1}$. Thus, if the activity starts at T_{1-k}, it finishes at $T_{1-k} + \vartheta_{1\text{-}1}$. Therefore, when $T_{1-k} < t_{1-K}$, if $t_{1-K} < T_{1-k} + \vartheta_{1\text{-}1}$ holds, R_1 would be entangled with the task of Route 1 at t_{1-K} such that R_1 cannot start to perform the sequence of Route 2 on time. Therefore, if $T_{1-k} \le t_{1-K}$ holds, $T_{1-k} < t_{1-K} - \vartheta_{1\text{-}1}$ should be satisfied. By (7), R_1 performs the sequence of Route 2 every $\vartheta_{2\text{-}1}$. If R_1 starts performing the sequence of Route 2 at t_{1-K}, the activity finishes at $t_{1-K}+\vartheta_{2\text{-}1}$. Therefore, when $t_{1-K} < T_{1-k}$, if $T_{1-k} < t_{1-K} + \vartheta_{1\text{-}1}$ holds, R_1 cannot start performing the sequence of Route 1 on time. Therefore, if $t_{1-K} \le T_{1-k}$ holds, $t_{1-K} < T_{1-k} - \vartheta_{2\text{-}1}$ should be satisfied. Then, the lemma holds. ■

Comparing with the schedule of Route 1, Route 2 works in relation to R_1 and R_2. Notice that, for Route 2, after a raw wafer is loaded into the BM by R_1, it is unloaded by R_2 soon. Meanwhile, by the swap strategy, a completed wafer is loaded into the BM by R_2 and unloaded from there by R_1. There is a strict order in the process. Therefore, it is important to schedule R_1 and R_2 at the BM so as to ensure Route 2 periodically work. Let t_{2-K} and t_{3-K} denote the time points that R_1 starts to perform $U_{b\text{-}1}$ and completes $L_{b\text{-}1}$ in Kth cycle for Route 2, respectively, and t_{4-K} and t_{5-K} are used to denote the time points that R_2 starts to perform $U_{b\text{-}2}$ and completes $L_{b\text{-}2}$ in Kth cycle for Route 2, $K\in\mathbf{N}_{\lambda_2}^+$. Then, we have

Lemma 3: Suppose that there is a process-bound twin-cluster tool. To ensure that Route 1 and 2 both are periodically scheduled in respective cycle time, the following conditions should be met.

$$t_{5-K} \le t_{2-K},\ K\in\mathbf{N}_{\lambda_2}^+ \tag{14}$$

$$t_{3-K} \le t_{4-K},\ K\in\mathbf{N}_{\lambda_2}^+ \tag{15}$$

Proof: Notice that for the BM of a twin-cluster tool, the wafer in it is raw or completed. After R_1 completes the swap sequence at the BM, a completed wafer is unloaded and a raw wafer is loaded into there. Therefore, before R_1 starts to perform $U_{b\text{-}1}$, R_2 should have loaded a processed wafer into BM, *i.e.*, (14) holds. After R_2 completes the swap sequence at BM, a raw wafer is unloaded and a completed wafer is loaded into there. Therefore, before R_2 starts to perform $U_{b\text{-}2}$, R_1 should have loaded a raw wafer into BM, *i.e.*, (15) holds. ■

Based on the above analysis, we propose a *Mathematical Programming Model* (MPM) to obtain a feasible periodic schedule by setting the robot waiting time. The objective of the MPM is to minimize the robot waiting time, and (16) represents the objective function. The MPM is as following.

$$max\ (w_{1-1} + w_{2-1} + w_{2-2}) \tag{16}$$

S. t.

$$(11) - (14)$$

$$T_{1-1} = 0 \tag{17}$$

$$t_{1-1} = T_{1-1} + \Lambda \tag{18}$$

$$T_{1-1} + \theta_{1\text{-}1} \le \Lambda \le T_{1-2} - \theta_{2\text{-}1} \tag{19}$$

$$T_{1-(k+1)} = T_{1-k} + \vartheta_{1\text{-}1},\ k\in\mathbf{N}_{\lambda_1}^+ \tag{20}$$

$$t_{1-(K+1)} = t_{1-K} + \vartheta_{2\text{-}2},\ K\in\mathbf{N}_{\lambda_2}^+ \tag{21}$$

$$t_{2-K} = t_{1-K} + v_1,\ K\in\mathbf{N}_{\lambda_2}^+ \tag{22}$$

$$t_{3-K} = t_{2-K} + v_1 + \alpha_1,\ K\in\mathbf{N}_{\lambda_2}^+ \tag{23}$$

$$t_{4-(K+1)} = t_{4-K} + \vartheta_{2\text{-}2},\ K\in\mathbf{N}_{\lambda_2}^+ \tag{24}$$

$$t_{5-K} = t_{4-K} + v_2 + \alpha_2,\ K\in\mathbf{N}_{\lambda_2}^+ \tag{25}$$

Algorithm 1: Calculate the robot waiting time such that two routes are periodically scheduled

Input: ρ_i, α_j, v_j, m ($i \in \mathbf{N}_2^+$)

Output: w_{1-1}, w_{2-1}, w_{2-2}, Λ

1. Initialization
   1.1 $\Pi_{ij} \leftarrow (\rho_i + 2 \times \alpha_j + v_j) / m_{ij}$, $i, j \in \mathbf{N}_2^+$
   1.2 $\Pi_j \leftarrow max\ (\Pi_{ij} \mid i \in \mathbf{N}_2^+), j \in \mathbf{N}_2^+$
   1.3 $\vartheta_{1\text{-}1} \leftarrow 6v_1 + 6\alpha_1 + w_{1-1}$
   1.4 $\vartheta_{2\text{-}1} \leftarrow 4v_1 + 4\alpha_1 + w_{2-1}$
   1.5 $\vartheta_{2\text{-}2} \leftarrow 6v_2 + 6\alpha_2 + w_{2-2}$
2. Calculate the robot waiting time by MPM

In (17), we assume that T_{1-k} serves as the reference time point for the subsequent time nodes. In (18), Λ represents the time interval between the T_{1-1} and t_{1-1}. Further, (19) describes the value range of Λ. Then, (20), (21), and (24) describe that the workings of Route 1 and 2 is periodic. Finally, (22), (23), and (25) describe the relationship of t_{1-K}, t_{2-K}, t_{3-K}, t_{4-K}, and t_{5-K}.

Based on the above analysis, Algorithm 1 is presented to find the best periodic schedule by calculating the minimal robot waiting time.

D. *Sufficient Conditions of the Optimal Scheduling*

We now analyze how a twin-cluster tool can be optimally scheduled. As shown in Fig. 4 (a), assume that $\Pi_1 = \Pi_2$ is satisfied, R_2 completes the swap sequence with the BM at time point t_1, R_2 starts to perform the swap sequence with BM again at time point t_2, and R_1 completes the task for Route 1 at time point t_3. Then, within the range [t_3, t_2], R_1 should load a raw wafer into the BM. Then, we have the following lemma.

Lemma 4: Suppose that there is a process-bound twin-cluster tool. If (26)–(28) are satisfied, Route 1 and 2 can both be optimally and periodically scheduled.

$$\Pi_1 = \Pi_2 \tag{26}$$

$$\Pi_1 \geq \theta_{1\text{-}1} + \theta_{2\text{-}1} \tag{27}$$

$$\Pi_2 - 2\alpha_2 - v_2 \geq \theta_{2\text{-}1} \tag{28}$$

Proof: Notice that if $\Pi_1 = \Pi_2 = \vartheta_{1\text{-}1} = \vartheta_{2\text{-}1}$ is satisfied, it means that in the cycle time $\vartheta_{1\text{-}1}$ ($\vartheta_{2\text{-}2}$), R_1 just need to perform the task for Route 1 and the task for Route 2 once, respectively. Correspondingly, R_2 need to perform the task for Route 2 once. Then, (27) enables that R_1 can perform the needed tasks in time, and (28) ensures that it is achievable for R_1 to perform the swap sequence at the BM during that BM contains a completed wafer, *i.e.*, $\vartheta_{1\text{-}1} = \Pi_1$ and $\vartheta_{2\text{-}1} = \Pi_2$ can be satisfied. Therefore, the lemma holds. ■

As shown in Fig. 4 (b), assume that $p \times \Pi_1 = \Pi_2$ is satisfied, $p \geq 2$, R_2 completes the swap sequence at the BM at time point t_1, and R_2 starts to perform the swap sequence with BM again at time point t_2. Within the range [t_1, t_2], R_1 should load a raw wafer into the BM. Then, we have the following lemma.

Lemma 5: Suppose that there is a process-bound twin-cluster tool. If (29) and (30) are satisfied, Route 1 and 2 can both be optimally and periodically scheduled.

$$p \times \Pi_1 = \Pi_2, p \geq 2 \tag{29}$$

$$\Pi_1 \geq \theta_{1\text{-}1} + \theta_{2\text{-}1} \tag{30}$$

Proof: Notice that if $p \times \Pi_1 = \Pi_2 = p \times \vartheta_{1\text{-}1} = \vartheta_{2\text{-}2}$ is satisfied, it means that in the cycle time $p \times \Pi_1$ (Π_2), R_1 just need to perform the task for Route 1 p times and the task for Route 2 one time, respectively, and R_2 need to perform the task for Route 2 one time. Then, (30) ensures that it is achievable for R_1 to load a raw wafer into the BM in time, *i.e.*, $\vartheta_{1\text{-}1} = \Pi_1$ and $\vartheta_{2\text{-}1} = \Pi_2$ can be satisfied. Therefore, the lemma holds. ■

As shown in Fig. 4 (c), assume that $\Pi_1 = p \times \Pi_2$ is satisfied, $p \geq 2$, R_2 completes the swap sequence at the BM at time point t_1, R_2 starts to perform the swap sequence at the BM again at time point t_2, and R_1 completes the task for Route 1 at time point t_3. For the tasks of Route 2, within the range [t_3, t_2], R_1 should load a raw wafer into the BM. Then, we have the following lemma.

Lemma 6: Suppose that there is a process-bound twin-cluster tool. If (31) and (32) are satisfied, Route 1 and 2 can both be optimally and periodically scheduled.

$$\Pi_1 = p \times \Pi_2, p \geq 2 \tag{31}$$

$$\Pi_2 \geq \theta_{1\text{-}1} + \theta_{2\text{-}1} \tag{32}$$

Proof: Notice that if $\Pi_1 = p \times \Pi_2 = \vartheta_{1\text{-}1} = p \times \vartheta_{2\text{-}2}$ is satisfied, $p \geq 2$, it means that in the cycle time $p \times \Pi_2$ (Π_1), R_1 should perform the task for Route 2 p times and the task for Route 1 one time, respectively, and R_2 should perform the task for Route 2 p times. (32) ensures that it is achievable for

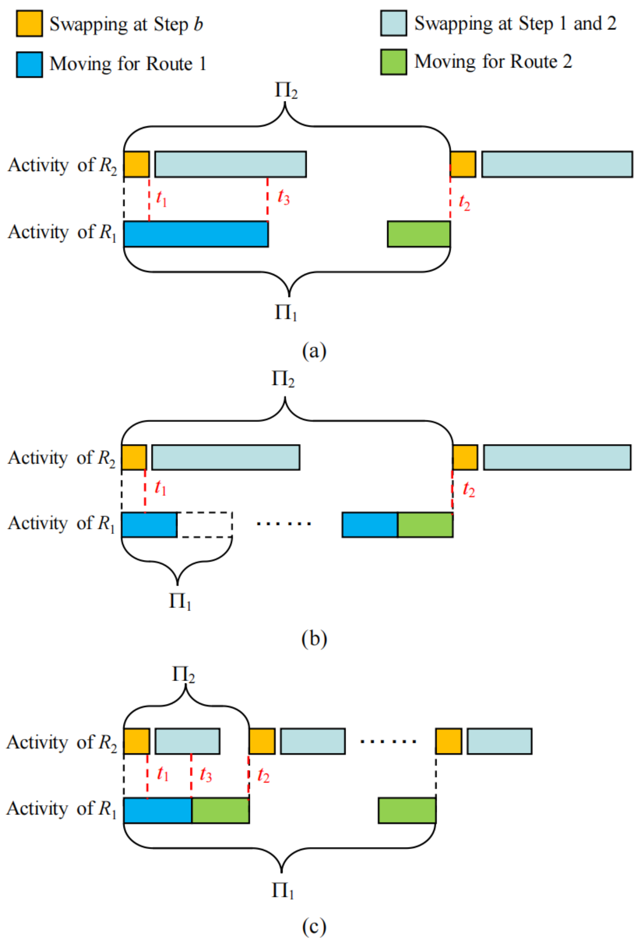

Figure 4. Gantt chart of robot task

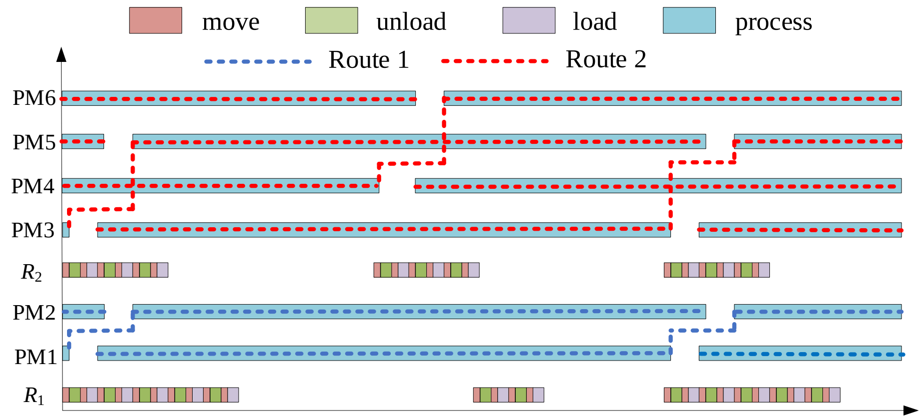

Figure 5. Gantt chart of Example 1

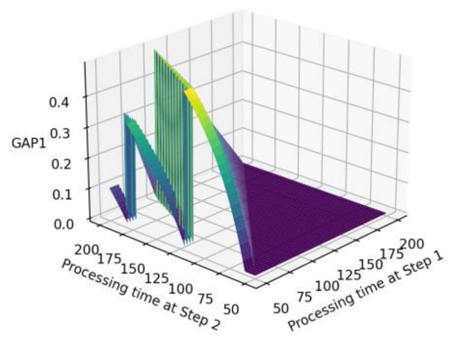

Figure 6. Performance of Algorithm 1 for CT1 as processing time increases

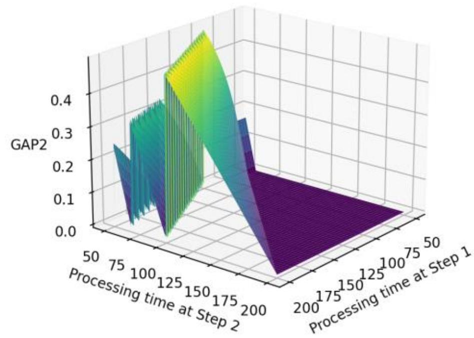

Figure 7. Performance of Algorithm 1 for CT2 as processing time increases

R_1 to complete the task for Route 2 in time, *i.e.*, $\vartheta_{1\text{-}1} = \Pi_1$ and $\vartheta_{2\text{-}1} = \Pi_2$ both can be satisfied. Therefore, the lemma holds. ■

III. EXPERIMENTS

An example is used to show the application of Algorithm 1 and the power it has.

Example 1: For a twin-cluster tool, in CT1, PM1 and PM2 are serving for Step 1 and Step 2, respectively; in CT2, PM3 and PM4 are serving for Step 1, and PM5 and PM6 are serving for Step 2 ($m = ([1,1], [2,2])$). A raw wafer should be processed at Step 1 for 200s, then be processed at Step 2 for 200s ($\rho_1 = 200$, $\rho_2 = 200$). For R_1, robot moving takes 1s, loading and unloading take 3s ($\alpha_1 = 3$, $\nu_1 = 1$). For R_2, robot moving takes 1s, loading and unloading take 3s ($\alpha_2 = 3$, $\nu_2 = 1$).

By (1), we can obtain that $\Pi_{11} = \Pi_{12} = (\rho_1 + 2\alpha_1 + \nu_1)/m_{11} = (200 + 2 \times 3 + 1) = 207$, $\Pi_{21} = \Pi_{22} = (200 + 2 \times 3 + 1)/2 = 103.5$. By (3) – (5), we have $\theta_{1\text{-}1} = 6\nu_1 + 6\alpha_1 = 24$, $\theta_{2\text{-}1} = 4\nu_1 + 4\alpha_1 = 18$, and $\theta_{2\text{-}2} = 6\nu_2 + 6\alpha_2 = 24$. Then, $\Pi_1 = max(\Pi_{11}, \Pi_{21}) = 207$ and $\Pi_2 = max(\Pi_{12}, \Pi_{22}) = 207$ can be obtained. By Algorithm 1, the maximum throughput of tool is 1/69, and robot waiting times can be obtained: $w_{1\text{–}1} = 183$, $w_{2\text{–}1} = 93$, and $w_{2\text{–}2} = 83$. Therefore, in this situation, the system reaches the maximum throughput. As shown in Fig. 5, Route 1 and 2 are running periodically with Algorithm 1.

Let GAPj describe the gap between Π_j and the cycle time obtained by Algorithm 1 of CTj, $j \in \mathbf{N}_2^+$, then can be obtained by following equations.

$$\text{GAP1} = \frac{\vartheta_{1\text{-}1} - \Pi_1}{\Pi_1} \tag{33}$$

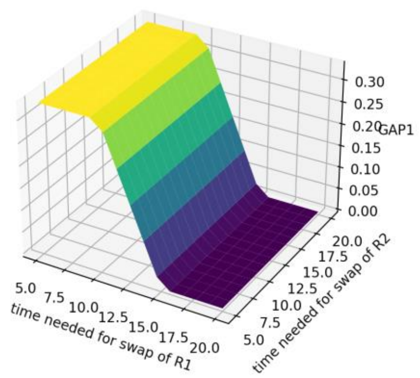

Figure 8. Performance of Algorithm 1 for CT1 as time needed for swap increases

$$\text{GAP2} = \frac{\vartheta_{2\text{-}2} - \Pi_2}{\Pi_2} \tag{34}$$

Fig. 6 and 7 display the variation of GAP1 and GAP2 as the processing time of each step increases. Assume that there is a twin-cluster tool with $\rho_1 \in [60, 200]$, $\rho_2 \in [60, 200]$, $\alpha_1 = 3$, $\nu_1 = 3$, $\alpha_2 = 3$, $\nu_2 = 2$, and $m = ([1, 2], [2, 1])$. Notice that, in most of cases, GAP1 and GAP2 tend to 0. Further, for some uncommon cases, GAP1 and GAP2 still are no more than 0.5.

We also analyze the relationship of robot action time and tool optimal schedule with Algorithm 1. Assume that there is a twin-cluster tool with WFP = [(2, 2), (2, 2)], $\rho_1 = \rho_2 = 160$, $(2\alpha_1 + \nu_1) \in [6, 20]$ and $(2\alpha_2 + \nu_2) \in [6, 20]$. According to Fig. 8 and 9, as the times needed for swap sequence of R_1 increase, GAP1 and GAP2 increase gradually. However, GAP1 and GAP2 also hold within an acceptable range.

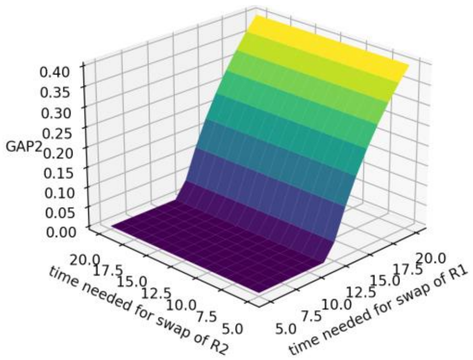

Figure 9. Performance of Algorithm 1 for CT2 as time needed for swap increases

As shown in Table II, there are six cases displayed to show the power of Lemma 4, 5, and 6, respectively. Notice that, for Cases 1, 3, and 5, they all are satisfy the demands of Lemma 4, 5, and 6, such that respective GAP1 and GAP2 both equal to 0. Further, for Cases 2, 4, and 6, they do not suit above sufficient conditions for optimal schedule such that, GAP1 or GAP2 is larger than 0.

TABLE II. PERFORMANCE OF SUFFICIENT CONDITIONS

No.	Parameters	Π_1	Π_2	$\vartheta_{1\text{-}1}$	$\vartheta_{2\text{-}2}$	GAP1	GAP2
1	$\rho_1 = 120$, $\rho_2 = 80$, $\alpha_1 = 5$, $v_1 = 2$, $\alpha_2 = 5$, $v_2 = 2$, WFP = ([2, 2], [2, 2])	66	66	66	66	0%	0%
2	$\rho_1 = 120$, $\rho_2 = 80$, $\alpha_1 = 3$, $v_1 = 2$, $\alpha_2 = 5$, $v_2 = 2$, WFP = ([2, 2], [2, 2])	64	66	66	66	3%	0%
3	$\rho_1 = 120$, $\rho_2 = 80$, $\alpha_1 = 5$, $v_1 = 2$, $\alpha_2 = 5$, $v_2 = 2$, WFP = ([1, 1], [2, 1])	132	66	132	66	0%	0%
4	$\rho_1 = 120$, $\rho_2 = 80$, $\alpha_1 = 3$, $v_1 = 2$, $\alpha_2 = 5$, $v_2 = 2$, WFP = ([1, 1], [2, 1])	128	66	132	66	3%	0%
5	$\rho_1 = 120$, $\rho_2 = 80$, $\alpha_1 = 5$, $v_1 = 2$, $\alpha_2 = 5$, $v_2 = 2$, WFP = ([1, 2], [1, 1])	66	132	66	132	0%	0%
6	$\rho_1 = 120$, $\rho_2 = 80$, $\alpha_1 = 5$, $v_1 = 2$, $\alpha_2 = 3$, $v_2 = 2$, WFP = ([2, 1], [1, 1])	66	128	66	132	0%	3%

IV. CONCLUSION

In recent years, twin-cluster tools have gained popularity due to their ability to reduce system load and improve space utilization efficiency through multiple processing routes. Meanwhile, periodic schedule is crucial for maintaining wafer quality. This paper focus on scheduling a twin-cluster tool to make sure that processing routes in the tool work periodically. By doing so, the workload of each module in the twin-cluster tool is analyzed to determine the optimal cycle time. Then, a mathematical programing model is presented to minimize the cycle time of the tool. An efficient algorithm is proposed to obtain the accurate robot task time. Further, several sufficient conditions are presented to ensure that the twin-cluster tool achieves the optimal cycle time. According to a number of effective examples, the effectiveness of presented method is confirmed. In practice, the wafer processing time of each chamber is a variable rather than a constant, which makes many existed studies unfeasible. In the future, we will analyze the system cycle time of twin-cluster tools with processing time variation.

REFERENCES

[1] D. -H. Roh, T. -G. Lee and T. -E. Lee, “K -Cyclic Schedules and the Worst-Case Wafer Delay in a Dual-Armed Cluster Tool,” IEEE Trans. Semicond. Manuf., vol. 32, pp. 236-249, May 2019.

[2] H. -J. Kim and J. -H. Lee, “Closed-Form Expressions on Lot Completion Time for Dual-Armed Cluster Tools with Parallel Processing Modules,” IEEE Trans. Automat. Sci. Eng., vol. 16, pp. 898-907, April 2019.

[3] W. Q. Xiong, C. R. Pan, Y. Qiao, N. Q. Wu, and M. X. Chen, “Reducing Wafer Delay Time by Robot Idle Time Regulation for Single-Arm Cluster Tools,” IEEE Trans. Automat. Sci. Eng., vol. 18, pp. 1653-1667, Oct. 2021.

[4] Y. Qiao, M. C. Zhou, N. Q. Wu, Z. W. Li and Q. H. Zhu, “Closing-Down Optimization for Single-Arm Cluster Tools Subject to Wafer Residency Time Constraints,” IEEE Trans. Syst., Man, Cybern., Syst., 51, pp. 6792-6807, Nov. 2021.

[5] C. R. Pan, M. C. Zhou, Y. Qiao and N. Q. Wu, “Modeling and scheduling of cluster tools dealing with wafer revisiting: A brief review,” in Proc. 12th IEEE Int. Conf. Net. Sen. Con. Taiwan, 2015, pp. 271-276.

[6] N. Q. Wu, C. B. Chu, F. Chu and M. C. Zhou, “A Petri Net Method for Schedulability and Scheduling Problems in Single-Arm Cluster Tools with Wafer Residency Time Constraints,” IEEE Trans. Semicond. Manuf., vol. 21, pp. 224-237, May 2008.

[7] T. -E. Lee, H. -Y. Lee, and Y. -H. Shin, “Workload balancing and scheduling of a single-armed cluster tool,” in Proc. 5th APIEMS Conf., Gold Coast, QLD, Australia, 2004, pp. 1–15.

[8] S. Venkatesh, R. Davenport, P. Foxhoven, and J. Nulman, “A steady-state throughput analysis of cluster tools: Dual-blade versus single-blade robots,” IEEE Trans. Semicond. Manuf., vol. 10, pp. 418–424, Nov. 1997.

[9] Q. H. Zhu, G. H. Wang, Y. Hou, N. Q. Wu and Y. Qiao, “Optimally Scheduling Dual-Arm Multi-Cluster Tools to Process Two Wafer Types,” IEEE Robot. Autom. Lett., vol. 7, pp. 5920-5927, July 2022.

[10] S. W. Ding, J. G. Yi, and M. T. Zhang, “Multi-cluster tools scheduling: an integrated event graph and network model approach,” IEEE Trans. Semicond. Manuf., vol. 19, pp. 339−351, Aug. 2006.

[11] J. G. Yi, S. W. Ding, D. Z. Song, and M. T. Zhang, “Steady-state throughput and scheduling analysis of multi-cluster tools for semiconductor manufacturing: a decomposition approach,” IEEE Trans. Automat. Sci. Eng., vol. 5, pp. 321−336, Apr. 2008.

[12] W. K. Chan, J. G. Yi, and S. W. Ding, “Optimal scheduling of multi-cluster tools with constant robot moving times, Part I: two-cluster Analysis,” IEEE Trans. Automat. Sci. Eng., vol. 8, pp. 5−16, Jan. 2011.

[13] Q. H. Zhu, N. Q. Wu, Y. Qiao, and M. C. Zhou, “Petri net-based optimal one-wafer scheduling of single-Arm multi-cluster tools in semiconductor manufacturing,” IEEE Trans. Semicond. Manuf., vol. 26, pp. 578−591, Nov. 2013.

[14] Q. H. Zhu, N. Q. Wu, Y. Qiao, and M. C. Zhou, “Scheduling of single-arm multi-cluster tools with wafer residency time constraints in semiconductor manufacturing,” IEEE Trans. Semicond. Manuf., vol. 28, pp. 117−125, Feb. 2015.

[15] Q. H Zhu, Y. Qiao and N. Q Wu, “Optimal integrated schedule of entire process of dual-blade multi-cluster tools from start-up to close-down,” IEEE/CAA J. Automat. Sinica, vol. 6, pp. 553-565, Mar. 2019.